EXPRESS MAIL CERTIFICATE

Date 7/26/02 Label No. EV115766025US

I hereby certify that, on the date indicated above, this paper or fee was deposited with the U.S. Postal Service & that it was addressed for delivery to the Assistant Commissioner for Patents, Washington, DC 20231 by "Express Mail Post Office to Addressee" service.

PATRICIA A. RUBIO
Name (Print)

Signature

PLEASE CHARGE ANY DEFICIENCY UP TO $300.00 OR CREDIT ANY EXCESS IN THE FEES DUE WITH THIS DOCUMENT TO OUR DEPOSIT ACCOUNT NO. 04 - 0100

Customer No.:

07278

PATENT TRADEMARK OFFICE

Docket No: 3598/0G116

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Sophie WILSON	Confirmation No.	5799
Serial No.:	09/395,297	Art Unit:	2183
Filed:	September 13, 1999	Examiner:	DECKTER, Stephanie M.

For: **CONDITIONAL INSTRUCTION EXECUTION IN A COMPUTER**

July 26, 2002

BOX FEE
Assistant Commissioner of Patents
Washington, DC 20231

MARK-UP FOR AMENDMENT OF JULY 26, 2002
PURSUANT TO 37 C.F.R. § 1.121

IN THE ABSTRACT:

"CONDITIONAL INSTRUCTION EXECUTION IN A COMPUTER"

--A METHOD AND SYSTEM FOR CONDITIONAL

INSTRUCTION EXECUTION IN A COMPUTER--.

A method and system for [computer system is disclosed capable of] conditionally carrying out an operation defined in a computer instruction[. The] wherein a computer instruction is implemented on so-called packed operands[, that is]; that is, operands containing a plurality of packed objects in respective lanes. An operation defined in the computer instruction is conditionally carried out in dependence on stored condition values which determine for each lane whether or not the operation is to be executed on objects in that lane.

[An execution unit for a computer system, a computer system and a method of executing instructions are defined.]

IN THE SPECIFICATION:

Page 3, delete the last paragraph and insert the new last paragraph as follows:

The computer system described herein [is schematically] , and illustrated in Figure 1, is a schematic diagram of the system. In Figure 1, reference numeral 2 denotes a program memory which holds programs in the form of a plurality of instructions. The program memory 2 is connected to an instruction cache 3 which is connected to instruction fetch/decode circuitry 4. The fetch/decode circuitry issues addresses to the program memory and received on each fetch operation a 64 bit instruction from the program memory 2 (or cache 3). Each 64 bit instruction can define two operations or a single operation. The decode unit 4 evaluates the opcode and transmits the appropriate control signals along X and Y channels 5_x,5_y. Each channel comprises a SIMD execution unit 8_x,8_y which includes three data processing units,

Page 5, delete the first full paragraph and insert the new first paragraph as follows:

processing unit. The destination address DST identifies a destination register into which a result of data processing will be placed. The operands and results are conveyed between the register file 10 or 11 and the respective data processing unit via the access paths, 12,14. In the case of load/store operations, the instruction formats allow memory access addresses A_x,A_y to be formulated from data values held in the registers as described in our copending U.S. Patent Application Serial No. 09/935,294 as described later. The load/store units access a common address space in the form of a data memory 16 via a dual ported data cache DCACHE 15. For this purpose, each load/store unit has a 64 bit data bus D_x,D_y and a 64 bit address bus A_x,A_y.

Page 11, delete the last paragraph and insert the new last paragraph as follows:

After comparing the test code specified in the addressed Treg byte with each of the condition codes CCX0 ... CCX7 (assuming the operation is being executed on the X side of the machine), then the specified operation is carried out on the SIMD lanes where there is a match, and is not carried out on the SIMD lanes where there is no match. An example is illustrated in Figure 7. Assumed that the operation illustrated in Figure 6 and described above has been carried out and that condition codes CCX0 to CCX7 have been set as described above depending on the results of the arithmetic operation in each of the SIMD lanes b_0 ...b_7. It is assumed for this example that the condition codes are $_{b0}$ 0010, b_1 0101, b_2 0011, b_3 0010, b_4 0010 b_5 0100. This is illustrated in the condition code register in Figure 7. Let us also assume that the addressed test register byte in the TST field of the instruction holds the condition code 0011. This denotes the condition Carry Set C. SIMD [lanes b_0,b_2,b_3,b_4 satisfy] satisfies this condition. Assume that the subsequent operation to be carried out is also an ADD instruction operating on the byte packed contents of two source registers SRC1,SRC2 with the results to be loaded into a

destination register DST. Because a test register byte has been specified, the addition operation is only effected on the

IN THE CLAIMS:

Cancel claim 6.

1. (Amended) An execution unit for use in a computer system for conditionally carrying out an operation defined in a computer instruction, the execution unit comprising:

first and second input stores for holding respective first and second operands on which an operation defined in the instruction is to be carried out, wherein each store [defines a plurality of lanes each holding an object] holds a plurality of objects of a predetermined size, each object defining one of a plurality of lanes, a maximum number of lanes being determined by a smallest allowable predetermined object size;

a plurality of operators associated respectively with [the] said lanes for carrying out an operation specified in the instruction on objects in corresponding lanes of [the] said first and second [source operands] input stores;

a destination buffer for holding the results of the operation on a lane-by-lane basis; and

selecting means for determining for each lane in dependence on stored condition values whether or not the operation is to be executed on objects in that lane;

wherein a number of stored condition values corresponds to said maximum number of lanes in each of aid first and second input stores, a prior operation being operable to generate said condition values so that, when the

operand have less than the maximum number of lanes, two or more condition values are set to a same value so that each individual condition value is generated regardless of a degree of packing of the first and second source operands.

4. (Twice Amended) An execution unit according to claim 2, wherein the number of condition codes in said set corresponds to the maximum number of lanes in each of the first and second source operands.

11. (Twice Amended) A method of executing instructions on operands containing a plurality of packed objects, the method comprising:

accessing at least one source operand containing a plurality of packed objects in a plurality of [respective] lanes, each packed object having a predetermined size and defining one of said plurality of lanes, a maximum number of said lanes being determined by a smallest allowable predetermined object size;

accessing stored condition values to determine for each respective lane whether or not an operation defined in the instruction is to be implemented on that lane of the operand; and

carrying out the operation and updating a destination operand only in those lanes for which [the] a stored condition value indicates that the operation should be implemented;

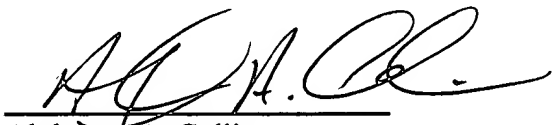
wherein a number of stored condition values corresponds to said maximum number of lanes in said at least one input store, a prior operation being

operable to set condition values so that, when the operands have less than a maximum number of lanes, two or more condition values are set to a same value so that each individual condition value is generated regardless of a degree of packing of the first and second source operands.

12. (Twice Amended) A method according to claim 11, wherein the stored condition values comprise a set of condition codes, held in a condition code register, and wherein the step of accessing the stored condition values comprises accessing [a] said set of condition codes [held in a condition code register] and comparing said [selected] condition codes with a test code identified in the instruction.

13. (Amended) A method according to claim 12, wherein the test code is held in a test register which is identified by an address in the instruction.

Respectfully submitted,



Alphonso A. Collins
Registration No. 43,559
Attorney for Applicant(s)

DARBY & DARBY
805 Third Avenue
New York, New York 10022
(212) 527-7700